# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

| | | |
|---|---|---|
| **1. AGENCY USE ONLY (Leave blank)** | **2. REPORT DATE**<br>31 August 2005 | **3. REPORT TYPE AND DATES COVERED**<br>Annual 1 Sep 04 – 31 Aug 05 |

**4. TITLE AND SUBTITLE**

CIP: A Complex Adaptive System Approach to QoS Assurance and Stateful Resource Management for Dependable Information Infrastructure

**5. FUNDING NUMBERS**

G - F49620-01-1-0317

**6. AUTHOR(S)**

Ye, Nong
Lai, Ying-Cheng Lai

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Arizona State University
Dept of Industrial Engineering
Box 875906
Tempe, AZ 85287-5906

**8. PERFORMING ORGANIZATION REPORT NUMBER**

XSA3285/TE

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

AFOSR
ATTN: Dr. Robert Herklotz /NL
4015 Wilson Boulevard, Room 713
Arlington, VA 22203-1954

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

AFRL-SR-AR-TR-07-0398

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**

APPROVED FOR PUBLIC RELEASE.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT *(Maximum 200 Words)***

This research relates to the project objectives concerning information infrastructure at the local, regional and global levels. The subgroup's major effort in the past year was on theoretical and computational issues concerning the structure, dynamics, optimization, information flow, and security in complex networks. The investigations directly relate to our original goals stated in previous years' reports. The benefits and applications of this network extend from small, local area networks, to large information infrastructures. Such networks may be governmental or civilian in nature. Critical government and civilian network operations can both benefit from having network QoS guarantees.

This year, we extended our local level quality of service (QoS) work on waiting time variance (WTV) problems and developed heuristic methods for weighted WTV problems. We drew some conclusions regarding the influencing factors of WTV. We investigated and began a hardware implementation of local level QoS solutions. At the regional level, we extended our local level QoS work to multiple machine problems. We completed investigating the case where the multiple machines are identical and developed methods for scheduling jobs on these machines. At the global level, we developed a protocol and algorithms for end-to-end QoS and began implanting a simulation to explore our protocol.

**14. SUBJECT TERMS** quality of service; QoS; waiting time variance

**15. NUMBER OF PAGES**

15

**16. PRICE CODE**

| | | | |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT**<br>UNCLASS | **18. SECURITY CLASSIFICATION OF THIS PAGE**<br>UNCLASS | **19. SECURITY CLASSIFICATION OF ABSTRACT**<br>UNCLASS | **20. LIMITATION OF ABSTRACT**<br>UNLIMITED |

# Annual Performance Report to AFOSR

| | |
|---|---|
| GRANT NO.: | F49620-01-1-0317 |
| PROJECT TITLE: | CIP: A Complex Adaptive System Approach to QoS Assurance and Stateful Resource Management for Dependable Information Infrastructure |
| PI: | Nong Ye |
| Co-PIs: | Ying-Cheng Lai |
| INSTITUTION: | Arizona State University<br>Tempe, Arizona |
| REPORTING PERIOD: | September 1, 2004 – August 31, 2005 |

## 1. Objectives

The objectives are unchanged from previous years.

## 2. Status of research

This research is directly related to the objectives of the AFOSR Project concerning information infrastructure at the local, regional and global levels. The investigations put forth here are directly in line with our original goals stated in previous years' reports. The benefits and applications of this network extend from small, local area networks, to large information infrastructures. Such networks may be governmental or civilian in nature. Critical government and civilian network operations can both benefit from having network QoS guarantees.

In this effort year we extend our local level quality of service (QoS) work on waiting time variance (WTV) problems and developed heuristic methods for weighted WTV problems. We also draw some conclusions regarding the influencing factors of WTV. We have investigated and began implementing a hardware implementation of our local level QoS solutions. At the regional level, we have extended our local level QoS work to multiple machine problems. We have completed investigating the case where the multiple machines are identical and developed methods for scheduling jobs on these machines. At the global level we have developed a protocol and algorithms for end-to-end QoS and began implanting a simulation to explore our protocol.

The subgroup's major effort in the past year was on theoretical and computational issues concerning the structure, dynamics, optimization, information flow, and security in complex networks. These are directly related to the Objectives of the AFOSR Project on information infrastructure at the regional and global levels.

## 3. Accomplishments/New Findings

Research highlights of work carried out by Dr. Ye, Dr. Lai and their post-doc and student assistants: During this reporting period, we accomplished the following:

- Developed heuristic methods for weighted WTV problems
- Made discoveries regarding the influencing factors of WTV
- Implemented our local level QoS work on a hardware router for feasibility testing
- Completed our work at the regional level on the identical parallel machine QoS problem
- Developed a protocol and algorithms for providing end-to-end QoS at the global level
- Gained an understanding of the mechanism of cascading breakdown in scale-free networks and developed practical strategies to prevent cascading breakdowns
- Investigated jamming in gradient complex networks
- Investigated traffic flow on complex networks

Elaboration on the above accomplishments follows:

### *3.1 Heuristic methods for weighted WTV problems*

The Weighted WTV (WWTV) minimization problem is to minimize the weighted waiting time variance of a batch of jobs on a single resource. We have described a mathematical formulation of the WWTV problem, analyzed the optimal sequences of several small-size WWTV problems and found a strong V-Shape tendency of the optimal sequences. In this section, we will first examine the optimal sequences by enumeration for some small-size WWTV problems. Based on the examination of optimal sequences, we proposed two methods to generate a job sequence that aims at reducing the weighted WTV of the jobs: Weighted Verified Spiral (WVS) algorithm and Weighted Simplified Spiral (WSS) algorithm.

*WVS*

Given a set of jobs $P = \{J_1, J_2, \ldots, J_n\}$ to process on a single resource, the processing time of job $J_i$ is $p_i$ and its weight is $v_i$ correspondingly, *i* from 1 to n.

1. To start, sort the job set $P$ to $P' = \{J'_1, J'_2, \ldots, J'_n\}$ such that $\frac{p_i^{'}}{v_i^{'}} \leq \frac{p_j^{'}}{v_j^{'}}$, $i \leq j$. The initial job sequence $\Omega$ has no jobs in it, and there are *n* jobs in the job pool $P'$ to be scheduled.
2. Remove jobs $J'_1$, $J'_{n-1}$, and $J'_n$ from the job pool $P'$ and put to the job sequence $\Omega$ as $\Omega = \{J_{n-1}^{'}, J_1^{'}, J_n^{'}\}$. Define sub sequences $R$ and $L$ such as $R = L = \Omega$. Job pool $P'$ becomes $\{J'_2, J'_3, \ldots, J'_{n-2}\}$.
3. Remove the job with the largest weighted processing time from the right side of the job pool $P'$. Try to place the job exactly before job $J'_1$ in sub sequence $R$ and calculate the weighted waiting time variance $WWTV_L$. Try to place the job exactly after job $J'_1$ in $R$ and calculate the weighted waiting time variance $WWTV_R$. If $WWTV_L$ is less than $WWTV_R$, let job sequence $\Omega = L$; otherwise $\Omega = R$. Update sub sequences L and R such as $R = L = \Omega$.
4. Repeat Step 3 until the job pool is empty and get the job sequence $\Omega$.

*WSS*

We notice that WVS needs to calculate WWTV of sub sequences R and L to decide the insertion position of each job in step 3 which adds the computational cost. Hence, we develop WSS, which needs less computation as follows:

1. To start, sort the job set P to $P' = \{J'_1, J'_2, \ldots, J'_n\}$ such that $\frac{p_i^{'}}{v_i^{'}} \leq \frac{p_j^{'}}{v_j^{'}}$, $i \leq j$. The initial job sequence $\Omega$ has no jobs in it, and there are n jobs in the job pool $P'$ to be scheduled. Define empty sub sequence L and R.
2. Remove the job with the largest weighted processing time from the right side of the job pool $P'$, insert it to the head of the sub sequence R.
3. Remove the job with the largest weighted processing time from the right side of the job pool $P'$, append it to the tail of the sub sequence L.
4. Repeat step 2 and 3 till the job pool is empty. The final job sequence $\Omega$ is the union of sub sequences L and R as $\Omega = L + R$.

Job sequence $\Omega$ by WSS has the structure as $\{J'_{n-1}, J'_{n-3}, \ldots, J'_{n-2}, J'_n\}$. We can see that WSS is more efficient than WVS with respect to the computational cost since it doesn't need to compute WWTV in each step.

We test the performance of WVS and WSS algorithms and compare them with First-In-First-Out (FIFO) and Weighed Shortest Processing Time first (WSPT). The testing results reveal that WVS and WSS methods are able to reduce WWTV compared with existing scheduling methods FIFO and WSPT. WSS can be applied to practical computers and network due to its computational efficiency and comparable performance to that of WVS.

### *3.2 Influencing factors of WTV*

In a previous study, we noticed that in addition to scheduling methods, the characteristics of the jobs, particularly the distribution of the processing times and sum of processing times (SOPT), impact WTV. Thus, we investigated the relationships among these factors and WTV. We found that there exists a quadratic relationship between the SOPT of a batch of jobs and WTV using 4 previously described scheduling methods: FIFO, SPT, Verified Spiral (VS) and Balanced Spiral (BS). We developed a mathematical formula to calculate the expected WTV for a batch of jobs whose processing times follow a given distribution. We discovered centralized mean WTV phenomena for normally and uniformly distributed problems using FIFO or SPT scheduling methods. We found a law of WTV variability that the problems with higher variation yield WTV with higher variation. We observed mean WTV shift phenomena that BS and VS scheduling methods are effective to reduce WTV compared to FIFO and SPT by producing smaller mean WTV.

Our findings of the factors influencing WTV are useful to systems administrators of computers and networks or anywhere WTV is concerned. With the understanding of the relationships between SOPT, distributions of the processing times of the jobs, scheduling methods and WTV, the system administrators could predict the mean WTV of the jobs, choose appropriate scheduling methods, and configure Admission Control schemes to achieve desirable WTV.

### *3.3 Local level implementation for feasibility testing*

To show the feasibility and performance of our local level QoS algorithms, we implement them on a research router using the Intel IXP1200 network processor. The router is able to run both algorithms for minimizing the WTV of jobs arriving for service. In initial experiments we process 1,000 packets in the router grouped in batches of size 10. Our results show that the WTV for a batch under the FCFS scheme with no admission control is 44,645. With BSAC added for admission control, and still using FIFO scheduling, we get a better WTV of 43,433. When we add BSAC and BS together, the WTV reduces even more to 36,770.

Thus, we find that our initial tests of implementing our methods on real hardware show that the algorithms are feasible and can improve performance with respect to minimizing job WTV. The details of our implementation, extended experimental results, and further analysis on performance metrics, such as WTV and running time, will be reported in future publications.

### *3.4 Identical parallel machine QoS problem*

Since the identical parallel machine CTV and WTV problems are NP-complete, the use of a heuristic algorithm for computational efficiency is justified. We developed five heuristic algorithms for the identical parallel machine WTV problem: FIFO+VS, SPT+VS, LPT+VS, DVS and DBS. We compare these five algorithms with FIFO, SPT and LPT alone. These 8 algorithms are described here.

1. FIFO (First-In-First-Out)
   FIFO is considered in this study because it is one of the most commonly used dispatching rules in scheduling and is widely used for a variety of Internet services. In FIFO, we assume the jobs arrive in a random order and all jobs have arrived. All machines are idle at the beginning. The first job is served by an idle machine. The next job will be served by another idle machine. If all machines are busy, then the next job will be served on a machine that becomes free next. That is, in FIFO both job dispatching to machines and job scheduling on each machine follow the FIFO order.
2. SPT (Shortest Processing Time)
   SPT is presented here because it is optimal to a related measure as presented in Pinedo, 1995. In the SPT heuristic, jobs are first sorted in increasing order of their processing times. The smallest *m* jobs are assigned to the first position on each machine, and then whenever a machine is freed, the next smallest job is assigned to that machine. That is, both job dispatching to machines, and scheduling on each machine, follow the order of SPT first.
3. LPT(Longest Processing Time)
   LPT is considered in this study because it is also optimal to a related measure. In LPT jobs are sorted in a decreasing order of their processing times. The largest *m* jobs will be assigned to the first position on each machine, and then whenever a machine is freed, the next largest job will be assigned to that machine. Hence, in LPT, job dispatching to machines and job scheduling on each machine follows the order of LPT first.
4. FIFO+VS (FIFO + Verified Spiral)
   FIFO+VS is shown in Figure 2.
5. SPT+VS (SPT + Verified Sprial)
   This is similar to FIFO + VS except that FIFO is replaced by SPT for job dispatching to machines in Step 1.
6. LPT+VS (LPT + Verified Sprial)
   This is similar to FIFO + VS except that FIFO is replaced by LPT for job dispatching to machines in Step 1.
7. DVS (Dynamic Verified Spiral)
   DVS checks and compares the waiting time variances from the possible assignments of a given job to a possible machine. The DVS heuristic is presented in Figure 3.
8. DBS (Dynamic Balance Spiral)
   DBS is similar to DVS except that VS is replaced by BS to schedule the jobs assigned to each machine $i \in M$. The BS method is shown in Figure 4.

We compare these heuristics for six small-size WTV problems, where WTVD is the Waiting Time Variance Deviation from the optimal solution and WTMD is the Waiting Time

Mean deviation from optimal. We find the optimal solution by enumerating all possible schedules. For each problem, FIFO, SPT and LPT are the worst among all the heuristics in waiting time variance. However, a significant improvement is made when VS is added to these three heuristics. DVS has the best performance in waiting time variance among all heuristics, and in 2 out of 6 problems it gives the optimal solution. The performance of DBS is very close to DVS, and DBS gives the optimal solution for one out of six problems.

We performed further testing on large size problems. The testing results showed that DVS gives the best performance in waiting time variance among all the heuristics for both small- and large-size problems. SPT+VS, LPT+VS and DBS also give good results in waiting time variance with significantly less computation complexity.

### *3.5 Protocol and algorithms for providing end-to-end QoS*

We first define and formalize the end-to-end QoS assurance problem. Next, we introduce a simulation design to investigate the application of our research work at the local and regional levels to this global QoS problem.

*Definition and formulation of the end-to-end QoS assurance problem*

We define end-to-end QoS assurance as a fixed path problem of self-interest only. Given the following, determine the arrival time of each flow at each hop along the path of the flow:

a. *n* flows
b. a fixed end-to-end path of each flow
c. end-to-end timeliness target of each flow
d. *m* resources required by all *n* flows
e. service capacity and waiting time at each resource from local-level and regional-level QoS models

We make the fixed path assumption based on existing evidences of a stable primary path of an end-to-end flow. For more than 50% of destinations there is only one dominant path, and for 25% of destinations there are exactly two domain-level paths (Govindan and Reddy, 1997). The reason for this is that routing policy is usually set by bi-lateral transit agreements. In most cases, a domain keeps a primary and a back-up transit to a collection of destinations. It is also shown that the likelihood of observing a dominant route is 82% at the host level, 97% at the city level and 100% at the autonomous system (AS) level (Paxson, 1996). In EGP (Exterior Gateway Protocol) for inter-AS routing, almost 90% of recorded updates contain close to 0% new information, indicating stable routes (Chinoy, 1993). The Border Gateway Protocol (BGP), contains only incremental updates. Injecting just 10% of the total inter-AS reachability information (about 200 entries) into the inter-AS routing permits the forwarding of at least 85% of the transit traffic without resorting to encapsulation (Rekhter and Chinoy, 1992) More than 80% of prefixes are reachable through a primary path for more than 95% of time.

In our ongoing work we consider subproblems of the fixed path problem of self-interest only. First, we allow each flow to determine its own arrival time by considering shortest possible time along with slack time. Next, we resolve timing conflicts at each resource while making the slack time of each flow ≥ zero. And finally, coordinate the resolutions at various resources. We also consider finding solutions for the open path problem. Such as assuring end-to-end QoS of some flows, expanding the set of *m* resources by pursuing alternative paths for those flows.

Subproblems here include selecting alternative resources for those flows and solving a fixed path problem with an expanded set of resources. For both fixed and open path problems, we consider self-interest and global interest of minimizing global traffic congestion (e.g., minimizing traffic bottlenecks).

### *A Job Reservation and Execution Protocol*

To address the end-to-end QoS assurance problem, we incorporate our work at the local and regional levels. Consider a network where variance in job waiting times is minimized. The processing time of a job depends on its size and can be calculated. By minimizing waiting time variance, we can predict the completion time (waiting time + processing time) of a job at each point in a network. Our network model considers only high priority jobs and assumes low priority jobs are handled whenever resources are idle.

In addition to incorporating our work at the local and regional levels, our experimental framework uses the concept of path reservation, as seen in the Resource ReSerVation Protocol (RSVP) proposal (Braden et al., 1997). However, we aim to overcome some problems with RSVP. We briefly overview RSVP and our protocol.

RSVP reserves a path for a flow on the Internet. This reservation is made at intermediate routers along the path. After a reservation for a flow is made, the jobs (in the form of a set of individual packets) in that flow travel along the reserved path. The idea behind this method is that by reserving resources to manage a flow, its QoS requirements can be assured. Some of the key points to RSVP are:

1. RSVP is receiver oriented, i.e., the receiver initiates the reservation request.
2. RSVP does not perform its own routing; it uses underlying routing protocols to determine where it should carry reservation requests, i.e., RSVP runs on top of the Internet Protocol.
3. Once the path is determined, the receiver sends a RESV packet with the bandwidth requirement along the determined path.
4. Each intermediate node on the path then makes a decision about accepting or rejecting the RESV request.
    a. Fail – NACK or error sent to the originator of the RESV packet.
    b. Success – set parameters in packet classifier and packet scheduler to achieve required QoS.

Our protocol is also based on path reservation and incorporates our work at the local and regional levels to minimize the variance of job waiting times at each point along the reserved path. We briefly outline the two phases (probe and job) of our protocol.

1. Probe Phase:
    a. At the source node, find the best path among *n* possible paths based on historic information and current state information. The source node subscribes to the historic performance and current state information from intermediate routers along *n* possible paths.

b. Source initiated: source sends a probe packet along the best path. The probe packet carries the parameters (job_id, start_time, $J$, $D_{ee}$), where $J$ is the job (packet) size, $D_{ee}$ is the end to end delay requirement.

c. Every intermediate router $i$ has three parameters ($B_{ij}$, $P_i$, $D_i$), where $B_{ij}$ is the size of batch $j$ at the router, $P_i$ is the processing power of the router, $D_i$ is the max (worst) possible delay that a packet can experience at this router. Each router also maintains a variable $B_{residual}$ that keeps track of how much resource is left at the router as reservations are made.

d. For each router $i$, upon receiving a probe packet:

```
If ($B_{residual} \geq J$),
    $D_{ee} = D_{ee} - D_i$
    If ($D_{ee}>0$)
            Forward probe packet
            $B_{residual}=B_{residual}-J$
            Add (job_id, probe_reply_timeout) to reservation list of batch j.
    Else
            drop probe packet
```

*(probe_reply_timeout = arrival time + timeout based on the avg roundtrip time)*

e. Upon destination receiving a probe packet:

```
If ($D_{ee}$–transportation time of the probe packet per job size unit * job size > 0)
    Return probe_reply packet
Else
    Drop probe packet
```

f. For each router $i$,

```
If (probe_reply packet not received by the time probe_reply_timeout)
    Drop the corresponding job from the list of jobs scheduled for batch j.
If (probe_reply packet arrives without a corresponding job_id in the list)
    Drop probe_reply packet
```

g. Source receives the probe_reply packet and enters Job Phase

2. Job Phase:

a. Source sends the job (packets) along the reserved path

b. Each intermediate router $i$, receiving the job checks to see if the job is in the job list for batch $j$

a. If yes

i. If the complete job arrives within its corresponding batch start time, schedule the job using BS

ii. If the job does not arrive before its corresponding batch's start time, drop the job from high priority queue

b. If no

i. Keep the job in the best-effort queue

From the brief outline given, observe that our protocol assumes admission control in batches. For this we use the BSAC method from our local and regional level work. During the job phase, jobs are scheduled using the BS algorithm from our local and regional level work to minimize the waiting time variance of jobs, which gives us the ability to determine the amount of time it will take a job to travel along a path, because we can compute its completion time at each router along the path (waiting time+processing time). Without stabilizing the waiting time variance, it would not be possible to make close predictions of the completion time at each point, thereby complicating the issue of maintaining timing along the path.

Our method uses a type of resource reservation that is different than RSVP. We compare our method to that of RSVP and list some of the main advantages of our method:

1. Non-Persistent reservation. RSVP reserves a path for an entire flow, we make the reservation for a specific job (set of packets) in a flow. Some advantages of this are:
    a. Resources are not wasted if the flow is not active. Consider the case of Voice over IP (VoIP) where there are distinct active and inactive periods. If we consider an active period as a job, then RSVP reserves the path for the duration of the call, whereas our protocol only reserves the path during the active periods.
    b. We target all types of applications, and the reservations are therefore valid only as long as they are needed irrespective of whether it's a short-term or a long-term connection.
    c. We may know all of the characteristics of a job, but not of a flow which has characteristics that may change over time.
2. Less State Info: Unlike RSVP, we store less state information. We store state information corresponding to two batches, the current batch and the next batch. The state information includes only job ids and the residual capacity of a batch.
3. Parallels Routing Algorithm: RSVP runs on top of IP. Our solution is integrated parallel to the routing algorithm to incorporate adjustments based on network dynamics. For example, a path that was good at the beginning of a VoIP session may at some point become congested. In our solution, a new path may be selected mid-session since path selection is done on a per job (active period) basis.
4. Light weight and Distributed: Our solution is light weight as it does not carry much state information and distributed as each router makes its own independent decision about accepting or rejecting a job.

One of the key benefits of our method is reducing resource wastage in a reservation. There are 3 ways to look at reserving paths. Reservation per packet, reservation per job (our method) and reservation per flow (RSVP). The first case, per packet, is clearly impractical as the reservation mechanism would significantly slow down network traffic. The last case, per flow, wastes too much resource along a path when a flow is inactive. We attempt to find a middle ground between the two by defining a job (set of packets) and making the reservation for that job.

Since we consider a network based on BSAC, we investigate batch sizes with respect to the number of packets in a job. The size of a batch is a variable that can be changed, adding flexibility to the framework. Another variable is the number of batches to reserve (persistence of reservation). For this, we consider the following optimization problem:

1. Let $C_{probe,}\, i$ be the cost of sending probe $i$
2. Let $C_{resv,}\, i$ be the cost of holding reservation for batch $i$
3. Let y be the optimum reservation persistance
4. Let $a_i$ be the number of packets in job $i$
5. Minimize $\sum (a_i / y) * C_{probe,i} + (y - a_i \% y) * C_{resv,i}$

6. s.t. $y>1$

Our proposed solution is currently in development. The ideas presented here are preliminary. We do not claim this as a complete solution for solving end-to-end QoS. However, we aim to provide a solution that is flexible, proactive, and secure. Flexibility is inherent in the variable parameters we allow (path selection, batch size, persistence of reservation) and the ability to change these parameters dynamically. As an example of proactive, consider sending packets in a flow, and stopping the flow after finding its QoS cannot be met (reactive), we do not release packets for a job until we know the QoS can be met (proactive). Our solution is more secure in that various "pieces" of a flow may not travel along the same path, thereby increasing the difficulty in eavesdropping. The use of BSAC and BS allows the overall benefits of time synchronization on a network, for which the implications are numerous.

In addition to the benefits we are continuing to explore, we also consider the tradeoffs. Obviously adding such synchronization and reservation to a network will consume resources and add processing time. Our ongoing efforts consider the advantages of our solution and weigh them against the shortcomings. We view this problem from a framework point of view, and are not currently actively trying to integrate it into existing protocols on the Internet.

### *3.6 Cascading breakdown in scale-free networks*

Complex networks arising in many natural and man-made systems are scale-free in that their connectivity (or degree) distributions follow an algebraic law. In such a network, a small subset of nodes can be significantly more important than others. From the standpoint of security, this means that the network can be fragile as attack on one or few nodes in this group can have a devastating effect. In particular, considering that those nodes typically handle a substantial fraction of loads necessary for the normal operation of the network, an attack to disable one or few of these nodes means that their loads will be redistributed to other nodes. Because the amount of the redistributed loads can be large, this can cause other nodes in the network to fail, if their loads exceed their capacities, which in turn causes more loads to be redistributed, and so on. This cascading process can continue until the network becomes totally disintegrated. Indeed, simulations show, for instance, that for a realistic power-grid network, attack on a single node can disable more than half of the nodes, essentially shutting down the network.

By utilizing a prototype cascading model, we previously determined the critical value of the capacity parameter below which the network can become disintegrated due to attack on a single node. A fundamental question in network security, which had not been addressed previously but may be more important and of wider interest, is how to design networks of finite capacity that are safe against cascading breakdown. We derived an upper bound for the capacity parameter, above which the network is immune to cascading breakdown. Our theory also yields estimates for the maximally achievable network integrity via controlled removal of a small set of low-degree nodes. The theoretical results are confirmed numerically.

Cascading breakdown of complex network can be catastrophic in a modern society. Our work represents a step toward understanding the dynamical mechanism of cascades and devising protective schemes in this important area of network security.

### *3.7 Jamming in gradient complex networks*

In networks such as the Internet, the financial-trade network, the neuronal system, the power-grid, metabolic network, etc., the flow properties of the transported entities (such as information, energy, chemicals, etc) become of primary interest. In particular, flow congestion, or jamming, and its dynamical relation to network structure has become a topic of recent investigation. The detailed mechanism for traffic flow varies from case to case, depending on the particular process in the network under consideration. For instance, in a neural network, flow of information is accomplished by the propagation and firing of electrical pulses. In the Internet, digital information flows according to a set of computer instructions. In a social network, rumor propagates along the routes established based on personal and/or professional relationships among individuals in the network. To be able to consider various networks in a general framework, it is reasonable to hypothesize the existence of a gradient field that governs the information flow on the network.

We investigated, analytically and numerically, under what conditions jamming in gradient flows can occur in random and scale-free networks. We found that the degree of jamming typically increases with the average connectivity of the network. A crossover phenomenon was uncovered where for relatively small average connectivity, scale-free networks have a higher level of congestion than random networks, while the opposite occurs for large connectivity. Our work indicated that the average network connectivity plays an important role in determining the susceptibility of scale-free networks to jamming as compared with random networks. For networks where the average connectivity is small, scale-free networks are more prone to jamming than random networks with the same average connectivity. Since most realistic networks have connectivities that fall in our ``small" regime, our result should be relevant. To ensure free information flow in complex networks is important and of broad interest to a variety of disciplines. Our results can be useful for understanding how jamming occurs and for devising strategies to minimize jamming in complex networks.

### *3.8 Onset of traffic congestion in complex networks*

Free, uncongested traffic flows on networks are critical for a modern society as its normal and efficient functioning relies on such networks as the internet, the power grid, and transportation networks, etc. To ensure free traffic flows on a complex network is naturally of great interest. One approach to addressing this important problem is modeling. Our particular interest is to understand under what conditions traffic congestions can occur on a complex network and to explore possible ways of control to alleviate the congestions. The models we have constructed are based on the setting of information transmission and exchange on the internet. There have been many previous works in this direction. A basic assumption used in these studies was that the network possesses a regular and homogeneous structure. Recent works have revealed, however, that many realistic networks including the internet are complex with scale-free and small world features. It is thus of paramount interest to study the effect of network topology on traffic flow, which is the key feature that distinguishes our work from the existing ones.

We developed two models for traffic flow on complex networks, taking into account the network topology, he information-generating rate, and the information-processing capacity of individual nodes. For each model, we studied four types of networks: scale-free, random, regular networks and Cayley trees. In the first model, the capacity of packet delivery of each node is proportional to its number of links, while in the second model, it's proportional to the number of shortest path passing through the node. We found, in both models, there is a critical rate of information generation, below which the network traffic is free but above which traffic congestions occur. Theoretical estimates were obtained for the critical point. For the first model, scale-free networks and random networks were found to be more tolerant to congestion. For the second model, the congestion condition is independent of network size and topology, suggesting that this model may be practically useful for designing communication protocols.

While our model was developed for computer networks, we expect it to be relevant to other practical networks in general, such as the postal service network or the airline transportation network. Our studies may be useful for designing new communication protocols for complex networks.

## 4. Personnel

***PI:***

Nong Ye
Professor of Industrial Engineering
Affiliated Professor of Computer Science and Engineering
Arizona State University

***Co-PI:***

Ying-Cheng Lai
Professor of Mathematics
Professor of Electrical Engineering
Arizona State University

***Post-doctoral fellow:***

Kwangho Park (10/1/03 - present)

***Graduate Research assistants:***

Xueping Li
Xiaoyun Xu
Anil Kumar Abraham
Narasimha Challa
Toni Farley

## 5. Publications

1. N. Ye, E. Gel, X. Li, and Y.-C. Lai, "Web-server QoS models: Applying scheduling rules from production planning." *Computers and Operations Research*, Vol. 32, No. 5, 2005, pp. 1147-1164.

2. N. Ye, “QoS-centric stateful resource management in information systems.” *Information Systems Frontiers*, Vol. 4, No. 2, 2002, pp. 149-160.
3. Z. Yang, N. Ye, and Y.-C. Lai, “QoS model of router with feedback control.” *Quality and Reliability Engineering International,* accepted.
4. N. Ye, X. Li, and T. Farley, “Job scheduling methods to reduce variance of job waiting times on computers and networks.” *Computers & Operations Research,* submitted.
5. X. Xu and N. Ye, “Minimization of job waiting time variance on identical parallel machines.” *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, submitted.
6. X. Li, N. Ye, and X. Xu, “Influencing factors of job waiting time variance for minimizing waiting time variance.” *IEEE Transactions on Systems, Man, and Cybernetics, Part A, submitted.*
7. X. Li, N. Ye, and X. Xu, “Job scheduling to minimize the weighted waiting time variance of jobs.” *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, submitted.
8. L. Zhao, Y.-C. Lai, K. Park, and N. Ye, ``Onset of traffic congestion in complex networks," Physical Review E 71, 026125 (2005).
9. N. Ye, Z. Yang, Y.-C. Lai, and T. Farley, “Enhancing router QoS through job scheduling with adjusted weighted shortest processing time - adjusted.” *Computers and Operations Research, Vol. 32, No. 9, 2005. pp. 2255-2269.*
10. T. Wu, N. Ye, and D. Zhang, “Comparison of distributed methods for resource allocation.” *Int'l Journal of Production Research, Vol. 43, No. 3, 2005, pp. 515-536.*
11. K. Park, Y.-C. Lai, and N. Ye, ``Characterization of weighted complex networks," Physical Review E, Vol. 70, No.2, 026109, 2004, pp. 1-4.
12. N. Ye, Y.-C. Lai, and T. Farley, “Dependable information infrastructures as complex adaptive systems.” *Systems Engineering,* Vol. 6, No. 4, 2003, pp. 225-237.
13. Y.-C. Lai, Z. Liu, and N. Ye, ``Infection dynamics on growing networks," International Journal of Modern Physics B 17, 4045-4061 (2003).
14. Y. Chen, T. Farley, and N. Ye, “QoS requirements of network applications on the Internet.” *Information, Knowledge, Systems Management*, Vol. 4, No.1, 2003, pp. 55-76.
15. Y.-C. Lai and N. Ye, "Recent developments in chaotic time series analysis," International Journal of Bifurcation and Chaos 13, 1383-1422 (2003).
16. Z. Liu, Y.-C. Lai, and N. Ye, "Propagation and immunization of infection on general networks with both homogenous and heterogeneous components," Physical Review E 67, 031911 (1-5) (2003). This work was selected by the Virtual Journal of Biological Physics Research for the April 1, 2003 issue (http://www.vjbio.org).
17. Z. Liu, Y.-C. Lai, N. Ye, and P. Dasgupta, "Connectivity distribution and attack tolerance of general networks with both preferential and random attachments," Physics Letters A 303, 337-344 (2002).
18. Z. Liu, Y.-C Lai, and N. Ye, “Statistical properties and attack tolerance of growing networks with algebraic preferential attachment.” Physical Review E. Vol. 66, 036112, 2002, pp. 1-7.
19. K. Park, L. Zhao, Y.-C. Lai, and N. Ye, ``Jamming in complex gradient networks," Physical Review E (Rapid Communications) 71, 065105 (2005)

20. L. Zhao, K. Park, and Y.-C. Lai, ``Attack vulnerability of scale-free networks due to cascading breakdown," Physical Review E (Rapid Communications), 70, 035101 (2004).
21. K. Park, Y.-C. Lai, and N. Ye, ``Self-organized scale-free networks," Physical Review Letters, in press.
22. L. Zhao, K. Park, Y.-C. Lai, and N. Ye, ``Tolerance of scale-free networks against attack-induced cascades," Physical Review E (Rapid Communications), in press.
23. K. Park, Y.-C. Lai, and N. Ye, ``Self-organized scale-free networks," Physical Review E, in press.
24. N. Ye, T. Farley, and D. Aswath, "Data Measures and Collection Points to Detect Traffic Condition Changes on Large-Scale Computer Networks." *Information, Knowledge, Systems Management,* accepted.
25. K. Park, Y.-C. Lai, L. Zhao, and N. Ye, ``Optimal network structure for efficient information transmission," in preparation.

## 6. Interactions/Transitions

Dr. Ye has maintained constant communications and research exchanges with the researchers at the Air Force Research Laboratory (AFRL), Rome, New York, including, John Faust, Patrick Hurley, and James Sidoran. John Faust and Patrick Hurley visited ASU in June 2005, and Dr. Ye visited AFRL in August 2005, along with frequent email communications among them for research exchanges and discussions of technology transfer.

Dr. Ye has also maintained a collaborative relationship with General Dynamics (Contact: Bruce Fette) and Symantec Corporation (Contact: Wesley Higaki) for technology transition purposes.

Participation/presentations at meetings, conferences, seminars, etc.:

Dr. Ye:

- "CIP: A Complex Adaptive System Approach to QoS Assurance and Stateful Resource Management for Dependable Information Infrastructure," The AFOSR Software & Systems Program PI Meeting, Rome, New York, August 16, 2005
- "Quality-of-Service Design of Computer/Network Systems for Dependability," Air Force Research Laboratory, Rome, New York, August 11, 2005
- "Quality of Service Design and Process Monitoring of Computer and Network Systems for Dependability and Security," Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, December 16, 2004

Dr. Lai:

- "Synchronization in complex networks," Joint Colloquium, Department of Mathematical Science and Department of Electrical Engineering, New Mexico State University, August 30, 2004.
- "Synchronization in complex networks," Joint Colloquium, Centre for Chaos Control and Synchronization and Department of Electronic Engineering, City University of Hong Kong, March 18, 2005.
- "Synchronization in complex networks," Seminar, Brazilian Institute for Space Research

(INPE), Sao dos Campos, Brazil, June 3, 2005.

- "Attack-induced cascades in complex networks: mechanism and prevention," Seminar, Department of Computer Science, University of Sao Paulo at Sao Carlos, Brazil, June 8, 2005.
- "Attack-induced cascades in complex networks: mechanism and prevention," Seminar, Department of Medicine, UCLA, August 4, 2005.

### 7. New discoveries, inventions, or patent disclosures.

One patent application by ASU:

- N. Ye, X. Li, T. Farley and H. Bashettihalli. Job Scheduling Techniques to Reduce the Variance of Waiting Time. US Patent Pending, ASU Case No. M3-069.

### 8. Honors/Awards

Dr. Ye is honored as 2003-present as an Adjunct Professor of Electronics and Computer Science at Peking University, Beijing, P. R. China, in 2003.

Society fellowship prior to this effort: Ying-Cheng Lai was elected as a Fellow of the American Physical Society in 1999 with the citation: For his many contributions to the fundamentals of nonlinear dynamics and chaos.